### Analysis of Database Practices: After Improvements

The refined database schema introduced critical fixes that enabled accurate, efficient analysis. First, a dedicated Products table was added, with proper fields for SKU, ProductName, and Price. This allowed product-related queries to use integer-based ProductID keys instead of parsing inconsistent text in the Status field. The Orders table was also normalized by introducing an OrderItems junction table, enabling one order to contain multiple products while maintaining referential integrity. The Status field was repurposed solely for tracking order progress (e.g., "Shipped"), eliminating confusion with product names.

Second, data integrity was enforced through foreign key constraints. The RMA table’s CustomerID column was set to NOT NULL, ensuring all returns could be traced back to customers. A ShippingAddress column was added to the RMA table, enabling direct regional analysis without relying on joins. State entries in the Customers table were standardized (e.g., always using "MA" instead of "Massachusetts"), preventing mismatches in geographic reports. Additionally, date validation rules were implemented to reject impossible values like 0000-00-00.

Third, performance was optimized through indexing and structured queries. Key columns (OrderID, CustomerID, ProductID) were indexed to speed up joins. Views were created for common reports (e.g., Monthly\_Return\_Rates), simplifying analysis. Stored procedures automated tasks like return rate calculations, reducing human error. The OrderItems table also allowed for accurate per-product metrics, such as calculating return rates by dividing RMA counts by OrderItems quantities rather than estimating with flawed Status filters.

Finally, reporting became more reliable and actionable. With clean, normalized data, queries could accurately identify top returned products, problem regions, and repeat returners. The separation of products from order statuses allowed for clearer dashboards, while standardized state codes ensured accurate geographic segmentation. Data validation rules prevented future inconsistencies, and the use of auto-incrementing primary keys eliminated duplicate entries. These improvements transformed the database from an unreliable repository into a robust system capable of driving business decisions.









